IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Long et al.	Attorney Docket No.: SUN1P280/P4889
Application No.: New	Examiner: Not Yet Assigned
Filed: Herewith	Group: Unknown

Title: METHOD AND APPARATUS FOR
EFFICIENTLY TRACKING MONITORS

PRELIMINARY AMENDMENT

Commissioner for Patents
BOX PATENT APPLICATION
Washington, D. C. 20231

Sir:

Prior to examination of the above application, please cancel claims 1 - 21 and add the following new claims.

22. A method for acquiring access to an object in a multi-threaded, object-based system, the method comprising:

associating a monitor with the object;

setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object; and

setting a reference substantially directly from the thread to the monitor so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector.

23. A method for acquiring access to an object as recited in claim 22 wherein setting the pointer and setting the reference occur substantially when the thread acquires a lock to the object.

24. A method for acquiring access to an object as recited in claim 22 wherein setting the reference includes setting a boolean reference count.

25. A method for acquiring access to an object as recited in claim 22 wherein associating the monitor with the object includes setting a first pointer from the monitor to the object.

26. A method for acquiring access to an object as recited in claim 22 wherein associating a monitor with the object includes obtaining the monitor from a freelist of monitors.

27. A method for acquiring access to an object as recited in claim 22 wherein associating the monitor with the object includes setting a second pointer from the object to the monitor.

28. A method for reducing overhead associated with providing a monitor for an object included in a multi-threaded, object-based computing system, the computing system including a plurality of monitors which includes the first monitor, the method comprising:

setting a first pointer from the first monitor to the object;

setting a second pointer from the object to the first monitor;

setting a third pointer from a thread to the object, wherein the thread is arranged to execute on the object; and

setting a reference substantially directly from the thread to the first monitor, wherein setting the reference includes updating contents of a reference field associated with the thread.

29. A method for reducing overhead as recited in claim 28 further including:

invoking the object using the thread, wherein invoking the object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state.

30. A method for reducing overhead as recited in claim 29 further including:

performing a memory reclamation during the wait state.

31. A method for reducing overhead as recited in claim 30 wherein performing a memory reclamation during the wait state includes:

identifying the contents of the reference field;

using the contents of the reference field to identify the first monitor; and

updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use.

32. A method for reducing overhead as recited in claim 31 wherein performing a memory reclamation during the wait state further includes:

scanning through substantially all monitors included in the plurality of monitors; and

reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is not reclaimed because the contents of the monitor field associated with the first monitor indicate that the first monitor is in use.

33. A computer program product for acquiring access to an object in a multi-threaded, object-based system, the computer program product comprising:

computer code for associating a monitor with the object;

computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;

computer code for setting a reference substantially directly from the thread to the monitor so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector; and

a computer-readable medium that stores the computer codes.

34. A computer program product according to claim 33 wherein the computer code for setting the reference includes computer code for setting a boolean reference count.

35. A computer program product according to claim 33 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.

36. A computer program product for reducing overhead associated with providing a monitor for an object included in a multi-threaded, object-based computing system, the computing system including a plurality of monitors which includes the first monitor, the computer program product comprising:

computer code for setting a first pointer from the first monitor to the object;

computer code for setting a second pointer from the object to the first monitor;

computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;

computer code for setting a reference substantially directly from the thread to the first monitor, wherein setting the reference include updating contents of a reference field associated with the thread; and

a computer-readable medium that stores the computer codes.

37. A computer program product according to claim 36 further including:

computer code for invoking the object using the thread, wherein invoking the object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state; and

computer code for performing a memory reclamation during the wait state.

38. A computer program product according to claim 37 wherein the computer code for performing a memory reclamation during the wait state includes:

computer code for identifying the contents of the reference field;

computer code for using the contents of the reference field to identify the first monitor;

computer code for updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use

computer code for scanning through substantially all monitors included in the plurality of monitors; and

computer code for reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is not reclaimed because the contents of the monitor field associated with the first monitor indicate that the first monitor is in use.

39. A computer program product according to claim 36 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.

40. A multi-threaded, object-based computing system comprising:

an object;

a thread that includes a first reference to the object; and

a monitor that includes a second reference to the object, wherein the object includes a third reference to the monitor and the thread includes a fourth reference to the monitor so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector.

Respectfully submitted,
BEYER, WEAVER & THOMAS, LLP

Michael J. Ferrazano
Registration No. 44,105

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, CA 94704
Tel: (650) 961-8300